Energy, Mines and Resources Canada

Énergie, Mines et Ressources Canada

CANMET

Canada Centre for Mineral and Energy Technology

Centre canadien de la technologie des minéraux et de l'énergie

Mining Research Laboratories

Laboratoires de recherche minière

Canada

ELLIOT LAKE LABORATORY X-RAY DIFFRACTION CONTROL AND ANALYSIS COMPUTER PROGRAM

M.G. Grenier and G. Knight
ELLIOT LAKE LABORATORY DIVISION REPORT MRL 89-32(TR)

January 1989

42 pp

MICROMEDIA

ELLIOT LAKE LABORATORY X-RAY DIFFRACTION CONTROL AND ANALYSIS COMPUTER PROGRAM

M.G. Grenier* and G. Knight**

ABSTRACT

This report describes the program XRD.BAS presently used by the Elliot Lake Laboratory to control the X-ray diffractometer. This apparatus is used to perform analysis of mine airborne dust samples for silica. The program is an updated version of the FORTRAN/SABR routine, originally developed by G. Knight. XRD.BAS is written in Microsoft Quickbasic and is intended for use with IBM compatible microcomputers.

Key words: X-ray diffraction; Silica; Computer control.

*Research Scientist; Elliot Lake Laboratory, CANMET, Energy, Mines and Resources Canada, Elliot Lake, Ontario. **Research Scientist, Mining Research Laboratories, CANMET, Energy, Mines and Resources Canada, Ottawa, Ontario.

PROGRAMME INFORMATIQUE DE COMMANDE DU DIFFRACTOMETRE DE DOSAGE A RAYONS X DU LABORATOIRE D'ELLIOT LAKE

M.G. Grenier* et G. Knight**

RESUME

Le présent rapport décrit le programme XRD.BAS qui est actuellement utilisé par le Laboratoired'Elliot Lake pour commander le diffractomètre à rayons X. L'appareil sert à doser la silice dans des échantillons de poussière aérienne de mine. Le programme est une version à jour du programme FORTRAN/SABR, initialement mis au point par G. Knight. Le programme XRD.BAS est rédigé en Microsoft Quickbasic et est destiné à être utilisé sur des microordinateurs compatibles IBM.

Mots clés: diffraction des rayons X; silice; commande par ordinateur.

*Chercheur scientifique; Laboratoire d'Elliot Lake, CANMET, Energie, Mines et Ressources, Elliot Lake (Ontario). **Chercheur scientifique, Laboratoires de recherche minière, CANMET, Energie, Mines et Ressources Canada, Ottawa (Ontario).

CONTENTS

	Page
ABSTRACT	i
RESUME	ii
INTRODUCTION	1
SOFTWARE DESCRIPTION	1
DATA FILES AND ADDITIONAL SOFTWARE	5
REFERENCES	6
APPENDIX A	A-8
APPENDIX B	B-31

INTRODUCTION

In early 1970, the Elliot Lake Laboratory developed a silica assessment method based on X-ray diffraction principles (1). Later the diffractometer and associated apparatus were coupled to a PDP8 computer from Digital in order to automatically process and analyze large sample batches. The original program is described at length elsewhere (2), and only a brief description of the changes made will be given here. Further reading is required in order to fully appreciate the programming code given in this report (3).

With time, the PDP8 developed serious electronic problems and steps were taken to replace the unit with an IBM compatible microcomputer. The programs were rewritten in Microsoft Quickbasic which is compiled as opposed to interpreted, thus allowing for faster running times. Quickbasic also has desirable features such as multi-line IF-THEN-ELSE statements and subprograms for easy modular programming.

SOFTWARE DESCRIPTION

The XRD.BAS program is listed in Appendix A. The following sections are a brief description of the program, subprograms and subroutines found in the code.

XRD.BAS MAIN PROGRAM

The main program begins by setting the base address for the four 24-bit parallel digital I/O interface cards. These cards are used to:

1. Control a switch box which provides the user with external controls while the apparatus is running. For example analysis may be stopped at any time and a final table printed out in case the diffractometer needs to be shut down. This switch register and its uses are described in detail elsewhere (4).

2. Acquire data in the form of counts from the diffractometer.
3. Supply the diffractometer with goniometer angle settings.
4. Automatically control the diffractometer and monitor its status. This includes controlling the scaler and the goniometer as well as changing samples.

The program then accesses subprogram XRDIOS where input parameters and sample data are entered. This includes instrument and sample filter types. number of samples as well as the number and order of mineral diffraction lines to be analyzed. Sample identification numbers and masses are also entered at this point.

The quartz standard is then pushed into the diffraction chamber and four quartz lines are analyzed (Miller indices (1,0,0), (1,0,1), (1,1,2). (2,1,1)). Following a successful standard run the samples are analyzed in turn. Quartz masses of individual samples are determined by comparing sample quartz peak areas to areas calculated from the standard run.

NXTSA SUBROUTINE

This section of the code pulls out the present sample, lowers the sample sled by one space and pushes in the next sample. Execution is suspended until the DSCBS (sample changer busy) signal is sent high indicating that the next sample is ready for analysis.

CNTCAL SUBROUTINE

This subroutine calculates the area and height of the quartz peak if one is detected.

SAMCAL SUBROUTINE

This part of the code sets the counting time. calculates the quartz mass from peak areas and standard calibration factors. It also checks for

agreement between quartz lines to allow shorter analysis times. If two quartz lines show similar masses. then the average of these two masses is accepted as the quartz mass for this sample and the next sample is processed. Otherwise, the next quartz line is analyzed. After all four lines have been run on a given sample, if none of the four masses calculated are in agreement, the program will proceed to print all pertinent data and plot the four associated quartz peaks. The sample will then be re-analyzed. If no match is found on the second run. the operator is informed that there probably is mineral interference on the sample.

DIAG SUBROUTINE

This section provides output data for all four quartz lines if no agreement was found during a given sample analysis.

XRDLIN SUBPROGRAM

This subprogram uses its own code as well as subroutines MAXVAL. PKEDGE. IFCNT. and DRIVE to obtain up to 29 counts required to define the quart peak for any given diffraction line. The subprogram first performs a series of counts on either side of the expected peak position. These counts are examined to establish the presence of quartz. If quartz is detected (positive curvature) the analysis continues by performing several counts on either side of the previously acquired counts. Once the peak edges have been detected, counting stops and the counts are passed on to subroutine CNTCAL where the peak area is calculated.

MAXVAL SUBROUTINE

This portion of the code finds the largest count obtained to date for a given quartz line. This is used to find the quartz peak location over a sloping background.

PKEDGE SUBROUTINE

This subroutine finds the beginning of the peak or peak edge.

IFCNT SUBROUTINE

This section is used to acquire a count via subroutine DRIVE if the required count has not been performed to data.

DRIVE SUBROUTINE

This routine gets the next angle from the main program, sends the goniometer arm to this angle, starts the counting process, times the count length, stops counting and sends the count value to XRDLIN.

XRDIOS SUBPROGRAM

This subprogram is used for input and output purposes. It gets input constants required for analysis from file XRD1.DAT and XRD2.DAT. These data files are described later on. It then allows the user to enter pertinent data such as instrument and filter type, number of samples and sample masses. The subprogram also provides an output for the results of analysis. A variety of print out formats are also available if interference is detected during analysis.

LDATA SUBROUTINE

This subroutine is used as a diagnostic tool or when analysis is inconclusive. Individual counts are printed out for each line.

SAMRES SUBROUTINE

This section is used for normal sample output. It prints out sample numbers, total respirable dust and respirable silica masses as well as other pertinent data.

PLOT SUBROUTINE

This subroutine plots the peaks for visual inspection. It is used in conjunction with LDATA when no match is found after all four lines have been analyzed.

FINTAB SUBROUTINE

This routine lists, in Table form, the results of analysis.

ABSORP SUBROUTINE

This routine is used to correct for absorption losses in the dust layer.

DATA FILES AND ADDITIONAL SOFTWARE

XRD1.DAT DATA FILE

This file contains constants used during the analysis. Factors are stored in XRD1.DAT by using the data preparation program XRDDATPR.BAS listed in Appendix B. These constants will be described later on along with the program.

XRD2.DAT DATA FILE

This file stores the standard areas calculated during the last run of XRD.BAS. These values are then used for comparing areas of the standard on a day to day basis. This allows the user to verify the validity of the present run and also helps monitor the X-ray tube wear on a long term basis.

XRD3.DAT DATA FILE

This file stores sample numbers, masses and data pertinent to any given run. It is updated every time XRD.BAS is run. This file is used to re-run a batch without having to re-enter individual sample data in the event of apparatus failure or operator error.

XRDDATPR.BAS PROGRAM

This program is used to examine or modify the XRD1.DAT data file used by XRD.BAS. The program is operated by menu and allows the user to perform changes easily. The data stored in XRD1.DAT consists of the following:

1. Four constants, two of which are meant to be used in conjunction with the automatic sample changer. The third one is a correction factor for the X-ray beam and filter disc plane offset. The fourth constant is the goniometer angle increment (typically 0.1 degree).
2. Twenty start-off angles for the goniometer. Angles 1 to 4 are used for the quartz lines, angles 5 to 20 are for performing other analysis such as crystobalite, tridymite and silver.
3. Four quartz masses for the standard lines analysis.
4. Sixteen factors for comparison with the quartz standard when performing crystobalite, tridymite or silver analysis.
5. Fifteen sets of four calibration factors for the various types of samplers that the X-ray diffraction apparatus can accommodate.
6. Twenty factors used to choose which standard quartz line is to be used for comparison when performing analysis of other minerals or silver.

REFERENCES

1. Knight, G., "CAMPEDS - A dust sampling and assessment system for use in hard rock mines": Ann Occup Hyg, vol. 30, No. 2, pp. 139-151: 1986.

2. Knight, G., "Quartz analysis X-ray diffraction with computer control - computer programs and explanations": Division Report MRL 88-8(R): CANMET, Energy, Mines and Resources Canada: 1988.

3. Knight, G., "Rapid XRD analysis of quartz in airborne dust samples using computer control": Ann Occup Hyg, in press; 1988.

4. Grenier, M.G., Ropchan, W. and Butler, K., "Elliot Lake Laboratory X-ray diffractometer IBM compatible controller and interface - technical information manual": Division Report MRL 89- (INT) (in preparation): CANMET, Energy, Mines and Resources Canada: 1989.

APPENDIX A

This appendix lists the Microsoft Quickbasic code for the program XRD.BAS.

```
REM  XRD program used to control x-ray diffraction apparatus and
REM  analysis.

     OPTION BASE 1

     DIM AREA(4),CONST(4),ANG(20),PEAK(7),COUNT(29),VAR3(4),EQUIVQ(20)
     DIM STDQM(20),QMASSA(12),QMASSH(12),MASSOUT(4,34),LINEOUT(12)
     DIM TIMEOUT(12),PEAKOUT(12),LIN(12),HEIGHT(94)

     COMMON SHARED AREA(),CONST(),ANG(),PEAK(),COUNT(),VAR1,TIME,VAR2
     COMMON SHARED EQUIVQ(),STDQM(),QMASSA(),QMASSH(),MASSOUT()
     COMMON SHARED TIMEOUT(),PEAKOUT(),LIN(),VAR3(),BGND1,SAMPNO
     COMMON SHARED RUNLIN,LMC,BGNDL,BGNDR,HEIGHT,AREA,PROB,PWIDTH
     COMMON SHARED BGND,LINEOUT(),SAMPTOT,TOTLIN

REM  CONFIGURATION OF 24 BIT PARALLEL DIGITAL I/O INTERFACE CARDS

REM  SWITCH REGISTER CARD AT BASE ADDRESS LOCATION 300

     FOR I1=1 TO 30:OUT &H303, &H9B:NEXT I1

REM  XRD COUNTS CARD AT BASE ADDRESS LOCATION 304

     FOR I1=1 TO 30:OUT &H307, &H9B:NEXT I1

REM  GONIOMETER ANGLE CARD AT BASE ADDRESS LOCATION 308

     FOR I1=1 TO 30:OUT &H30B, &H80:NEXT I1

REM  CONTROL COMMANDS CARD AT BASE ADDRESS LOCATION 30C

     FOR I1=1 TO 30:OUT &H30F, &H8B:NEXT I1

REM  Input subprogram called, sets parameters such as : instrument type
REM                                                     filter type
REM                                                     number of sample
REM                                                     lines to analyse
REM                                                     order of lines
REM
REM  Also allows sample numbers and masses to be entered.

     IO=1
     CALL Xrdios(IO)
```

```
REM  FILTERTYPE is set equal to VAR2 (filter membrane type)

     FILTERTYPE=VAR2

REM  Output arrays set to zero

     FOR J = 1 TO 12
         LINEOUT(J)=0
         TIMEOUT(J)=0
         PEAKOUT(J)=0
         QMASSH(J)=0
         QMASSA(J)=0
     NEXT J

REM  Call subprogram Nxtsa to push in next sample

     GOSUB Nxtsa

REM  Do standard analysis.

     SAMPNO=0

     FOR RUNLIN = 1 TO 4
         TIME = 10
         IF (RUNLIN - 2) = 0 THEN TIME = 9
         VAR1 = ANG(RUNLIN)

REM  Call subprogram Cntcal to do counting and calculating work on next
REM  sample.

         GOSUB Cntcal

REM  Call subprogram Xrdios to produce detailed line output for
REM  standard.

         IO=2
         CALL Xrdios(IO)
         HEIGHT(RUNLIN)=HEIGHT
         AREA(RUNLIN)=AREA
     NEXT RUNLIN

     LPRINT"-------------------------------------------------------------
     LPRINT"-----------------"
     LPRINT

REM  Do sample analysis for samples 1 to SAMPTOT

     FOR SAMPNO = 1 TO SAMPTOT
```

```
        FOR J = 1 TO 12
            LINEOUT(J)=0
            TIMEOUT(J)=0
            PEAKOUT(J)=0
            QMASSH(J)=0
            QMASSA(J)=0
        NEXT J

        GOSUB Nxtsa
        LMCFACT=1

REM  CHECK SW#1 FOR OMITTING LIGHT SAMPLE CORRECTION (SKIP IF SW UP).

        SW1=INP(&H300)
        IF (SW1 AND 1) = 1 THEN GOTO 10

REM  Do light sample correction for 5 and 3 micron pore size filters
REM  for samples less than .4 mg in mass.

        IF FILTERTYPE=0 OR FILTERTYPE=1 THEN LMC=1 ELSE LMC=0

        IF LMC = 1 THEN
            W1=MASSOUT(4,SAMPNO)

            IF (W1-.4) <= 0 THEN
                X=1.4-W1*(1.4-W1)

                IF (FILTERTYPE-1) = 0 THEN
                    LMCFACT=SQR(X)
                ELSEIF (FILTERTYPE-1) < 0 THEN
                    LMCFACT=X
                END IF

            END IF

        END IF

10      FOR J = 1 TO 4
            VAR3(J)=0
        NEXT J

        MASSOUT(3,SAMPNO) = -1000

REM  CHECK SW#2 TO RUN ALL LINES OR STOP ON MATCH (SW UP RUN ALL)

        SW2=INP(&H300)
        IF (SW2 AND 2) = 2 THEN L5=200 ELSE L5=0
        L=0:L8=0:J2=0
20      L8 = L8 + L
```

```
REM  Check for last lined analyzed

     IF (L8 - TOTLIN) > 0 THEN GOTO 70

REM  Diff. lines processed in batches (J2) of 4 lines (quartz, cryst...)

         J2 = J2 + 1

30       FOR L = 1 TO 4
             L9 = L + L8
             RUNLIN = LIN(L9)
             L7 = RUNLIN - J2 * 4
             IF (L9 - TOTLIN) > 0 THEN GOTO 40
             IF L7 > 0 THEN GOTO 50
             IF L5 < 0 THEN GOTO 40

             GOSUB Samcal

             IF (RUNLIN - 20) = 0 THEN MASSOUT(3,SAMPNO) = W
             IF (L5 - 100) <= 0 THEN GOTO 40
             IF (VAR2 - 1) > 0 THEN GOTO 40

             IO=3
             CALL Xrdios(IO)

40       NEXT L

         L = 5

REM  Go to next sample if QMASSH's are set to -1003 (start of Cntcal)

50       X=QMASSH(1) + (2) + 2006
         IF X <= 0 THEN GOTO 70

REM  Diagnostic control

         L = L - 1

         IF L5 > 0 THEN
             IF (L5 - 200) <= 0 THEN GOTO 20 ELSE GOTO 60
         ELSEIF L5 = 0 THEN

             IF (J2 - 2) > 0 THEN
                 GOTO 20
             ELSEIF (J2 - 2) = 0 THEN
                 IF L <= 0 THEN GOTO 20
60               L5 = 0
                 GOTO 20
```

```
            ELSEIF (J2 - 2) < 0 THEN
                L5 = 300
                IO=5
                CALL Xrdios(IO)
                GOSUB Diag
                GOTO 30
            END IF

        ELSEIF L5 < 0 THEN
            GOTO 60
        END IF

70      IO=5
        CALL Xrdios(IO)

REM  CHECK SW#3 FOR OUTPUTTING OF FINAL TABLE AFTER THIS SAMPLE
REM  IF SW#3 IS UP THEN END.

        SW3=INP(&H300)
        IF (SW3 AND 4) = 4 THEN IO=6 : CALL Xrdios(IO) : END
    NEXT SAMPNO

    SAMPNO=SAMPTOT
    IO=6
    CALL Xrdios(IO)
    END

REM  Subroutine Nxtsa changes sample automatically.

    Nxtsa:

REM  Set output command lines low.

    FOR I1=1 TO 30:OUT &H30C, &H0:NEXT I1

REM  Activate ASC.

    FOR I1=1 TO 30:OUT &H30C, &H40:NEXT I1:FOR FF=1 TO 50:NEXT FF
    FOR I1=1 TO 30:OUT &H30C, &H0:NEXT I1

REM  Push in next sample.

    FOR I1=1 TO 30:OUT &H30C, &H10:NEXT I1:FOR FF=1 TO 50:NEXT FF
    FOR I1=1 TO 30:OUT &H30C, &H0:NEXT I1

REM  Check DSCBS signal acknowledging end of sample change process.

    DSCBS=&H0
```

```
      WHILE (DSCBS AND &H4) <> &H4
          DSCBS=INP(&H30D)
          LOCATE 1,1:PRINT"CHANGING SAMPLE"
          FOR FF=1 TO 500:NEXT FF
      WEND

      CLS

      RETURN

REM   Subroutine Cntcal uses counts to calculate area and height.

      Cntcal:

      CALL Xrdlin
      N3 = PEAK(3)
      IF (VAR2-1) > 0 THEN QMASSH(L9)=-1000-I : IO=7 : CALL Xrdios(IO)
      IF VAR2 < 0 THEN AREA=0 : HEIGHT=0 : RETURN
      X=0 : Y=0 : W=0
      N1=PEAK(1) : N2=PEAK(2)
      N4=PEAK(4) : N5=PEAK(5)

      FOR N = N4 TO N5
          X=X+COUNT(N)
      NEXT N

      FOR N = N1 TO N2
          Y=Y+COUNT(N)
      NEXT N

      FOR N = N2 TO N4
          W=W+COUNT(N)
      NEXT N

      X1=N5-N4+1
      BGNDL=X/X1/VAR1
      X1=N2-N1+1
      BGNDR=Y/X1/VAR1
      X1=N4-N1
      X2=N5-N2
      X3=N4-N2+1
      X4=N5+N4-N2-N1
      AREA=W/VAR1-(BGNDL*X1+BGNDR*X2)*X3/X4
      BGND1=(BGNDL-BGNDR)/X4*2
      X1=N5-N3
      X2=N3-N1
      X3=N5-N1
      HEIGHT=COUNT(N3)/VAR1-(BGNDL*X1+BGNDR*X2)/X3

      RETURN
```

```
REM  Subroutine Samcal sets count time and checks line agreement.

     Samcal:

     TIME=20
     IF RUNLIN=2 OR RUNLIN=5 OR RUNLIN=20 THEN TIME=10
     VAR1=ANG(RUNLIN)+CONST(3)
     GOSUB Cntcal

REM  Set output code.

     L3=EQUIVQ(RUNLIN)
     LINEOUT(L9)=RUNLIN
     TIMEOUT(L9)=CINT(TIME*.4096)
     PEAKOUT(L9)=N3

REM  Calculate mass corresponding to peak area.

     X=STDQM(RUNLIN)*LMCFACT*1000
     W=AREA*X/AREA(L3)

REM  I in the following statement is returned from Subroutine
REM  Xrdlin to signify :    -1 no quartz
REM                          0 measurable quartz
REM                          1 possible interference

     IF VAR2 > 0 THEN
         IF (W+20) < 0 THEN W=-1000
         IF W < 0 THEN W = 0
     END IF

     QMASSH(L9)=HEIGHT*X/HEIGHT(L3)
     QMASSA(L9)=W
     IF (J2-2) > 0 THEN RETURN
     N=L-1

     IF N > 0 THEN

         FOR J=1 TO N
             Y=QMASSA(L8+J)
             X=(W+Y)*.06+20
             X=ABS(W-Y)-X
             IF X <= 0 THEN
                 VAR3(J2)=VAR3(J2)+2
                 VAR3(J2+2)=VAR3(J2+2)+W+Y
                 L5=L5-1
             END IF
         NEXT J

     END IF

     IF (L-4) >= 0 THEN RETURN
     J=L*30+4
```

```
      FOR N=1 TO 29
          HEIGHT(J-N)=COUNT(N)
      NEXT N

      HEIGHT(J)=VAR1

      RETURN

REM   Subroutine Diag outputs all data on quartz lines when no agreement
REM   is found or upon request.

      Diag:

      IO=3
      CALL Xrdios(IO)
      K1=L-1

      FOR K = 1 TO K1
          J=K*30+4

          FOR N = 1 TO 29
              COUNT(N)=HEIGHT(J-N)
          NEXT N

          TIME=HEIGHT(J)
          IO=4
          CALL Xrdios(IO)
      NEXT K

      RETURN

REM   Subprogram Xrdlin for diffraction line analysis.

      SUB Xrdlin STATIC

REM   DELETE THE 1 IN TIME=TIME*3+1 EVENTUALLY
REM   IN ORDER THAT THE FOLLOWING STATEMENT, WHICH IS
REM   DESIGNED TO STOP THE SCALER FROM EXCEEDING
REM   1000000 COUNTS ON THE SECOND QUARTZ LINE, MAY
REM   OPERATE PROPERLY.

      IF PROB=4 THEN
          TIME=TIME*3+1
          IF TIME = 27 THEN TIME=13
      END IF

      FOR N=1 TO 29
          COUNT(N)=0
      NEXT N

      I2=0
      UANG=VAR1
      Y=0
```

```
      FOR I=1 TO 2
          TIME=TIME+2*I2
          I2=I2+TIME
          VAR1=TIME
          N2=15-PWIDTH
          N3=15+PWIDTH

          FOR N= N2 TO N3
              GOSUB Drive
              Y=Y+W
              COUNT(N)=COUNT(N)+W
          NEXT N

          TIME=I2 : VAR1=TIME : X1=2*PWIDTH+1
          X2=COUNT(N3)+COUNT(N3-1)
          X3=COUNT(N2)+COUNT(N2+1)
          BGND=(X2-X3)/(2*X1-4)
          Y1=Y/X1
          X2=(X2+X3)/4
          X1=SQR(Y1)
          GOSUB Maxval
          IF (Y1-X2+X1) < 0 GOTO 90
          IF (X-Y1-8*X1) >= 0 GOTO 90
          IF (Y1-X2+PROB*X1/6) < 0 GOTO 80
          IF (Y1-X2-PROB*X1) >= 0 GOTO 90
      NEXT I

      IF (Y1-X2-.7*X1) > 0 GOTO 90

80    VAR2=-1
      EXIT SUB

90    IF (N1-3-N2) < 0 THEN N2=N1-3
      IF (N1+3-N3) > 0 THEN N3=N1+3

      GOSUB Maxval
      N=ABS(N1-15)

      IF (PWIDTH-N) < 0 THEN
          VAR2=1
          PEAK(1)=N2
          PEAK(2)=N2+1
          PEAK(4)=N3-1
          PEAK(5)=N3
          EXIT SUB
      END IF

      PEAK(6)=N1+8
      PEAK(7)=N1-8
      PEAK(1)=N1+3
      PEAK(2)=N1-3
```

```
FOR J=1 TO 7
    X3=PEAK(1)-PEAK(2)-1
    BGND=COUNT(PEAK(1))+COUNT(PEAK(1)-1)-COUNT(PEAK(2))
    BGND=(BGND-COUNT(PEAK(2)+1))/X3

    FOR I=1 TO 2
        I2=3-2*I
        X2=I2

        IF (PEAK(5+I)-8*I2-N1) = 0 THEN
            N=PEAK(I)
            W=COUNT(N)
            W=W+1.5*SQR(W)-COUNT(N-2*I2)-BGND*X2

            IF W < 0 THEN
                N=N+I2
                GOSUB Ifcnt
                PEAK(I)=N
            ELSEIF W >= 0 THEN
                PEAK(5+I)=PEAK(I)-2*I2
            END IF

        END IF

        N2=PEAK(6)
    NEXT I

    N3=PEAK(7)
NEXT J

X=3*(N2-N3)+6


BGND=COUNT(N2)+COUNT(N2+1)+COUNT(N2+2)-COUNT(N3)-COUNT(N3-1)
BGND=BGND-COUNT (N3-2)
BGND=BGND/X
GOSUB Maxval
N=ABS(N1-15)

IF (PWIDTH-N) < 0 THEN
    VAR2=1
    PEAK(1)=N2
    PEAK(2)=N2+1
    PEAK(4)=N3-1
    PEAK(5)=N3
    EXIT SUB
END IF
```

```
      FOR K=1 TO 2
          I2=3-2*K
          JJ2=INT((N2-N3)/5+3)
          N=PEAK(5+K)+I2
          GOSUB Pkedge
          PEAK(3+2*I2)=N+I2
          JJ2=ABS(N-N1)
          IF JJ2 <= 0 THEN JJ2=1
          I2=-I2
          GOSUB Pkedge
          PEAK(3-I2)=N+I2
      NEXT K

      BGND=BGND/VAR1
      VAR2=0

      EXIT SUB

REM   Maxval finds a peak above a sloping background.

      Maxval:

      X=0
      IF N2 <= N3 THEN ST = 1 ELSE ST = -1

      FOR N=N2 TO N3 STEP ST
          GOSUB Ifcnt
          X3=N-15
          W=W-BGND*X3
          IF (W-X) > 0 THEN  X=W : N1=N
          PEAK(3)=N1
      NEXT N

      RETURN

REM   Pkedge finds start of peak from background and shoulder.

      Pkedge:

      Y=0

      FOR J=1 TO JJ2
          X=I2
          N=N+I2
          J1=-2
          IF N < 1 OR N > 29 THEN GOTO 100
          X3=N
          GOSUB Ifcnt
          X1=J
          Y=Y+COUNT(N-2*I2)-(X3-2*X)*BGND
          X2=Y/X1+1.5*SQR(W)
```

```
        IF (W-X3*BGND-X2) > 0 THEN
            J1=-3
            IF (COUNT(N-I2)-(X3-X)*BGND-X2) >= 0 THEN GOTO 100
        END IF

        J1=J1+1
    NEXT J

100 N=N+J1*I2

    RETURN

REM Subroutine Ifcnt sends control to Drive if COUNT(n) is zero.

    Ifcnt:

    W=COUNT(N)
    IF W <= 0 THEN GOSUB Drive
    COUNT(N)=W

    RETURN

REM Subroutine Drive :  - converts angle value from binary to BCD
REM                     - initiates goniometer movement
REM                     - resets scalar
REM                     - starts, times, stops and reads counts
REM                     - converts counts from BCD to binary

    Drive:

REM CHECK SW#4 FOR END ANLYSIS NOW, IF SW#4 IS UP END NOW.

    SW4=INP(&H300)
    IF (SW4 AND &H8) = &H8 THEN IO=6 : CALL Xrdios(IO) : END

REM Use angle from main program  and send to angle registers.

    X1=N-15
    ANG=(UANG+CONST(4)*X1):ANG=CINT(ANG*100):ANG$=STR$(ANG/100)

    IF LEN(ANG$)=7 THEN

        FOR I1=1 TO 30:OUT &H30A,&H1:NEXT I1

    ELSE

        FOR I1=1 TO 30:OUT &H30A,&H0:NEXT I1

    END IF
```

```
      ANG$=RIGHT$(ANG$,5)
      P91=VAL(LEFT$(ANG$,1))
      P92=VAL(MID$(ANG$,2,1))
      P81=VAL(MID$(ANG$,4,1))
      P82=VAL(RIGHT$(ANG$,1))
      P309=P91*16+P92:P308=P81*16+P82
      FOR I1=1 TO 30:OUT &H308,P308:NEXT I1
      FOR I1=1 TO 30:OUT &H309,P309:NEXT I1

REM   Send signal to start goniometer arm movement.

      FOR I1=1 TO 30:OUT &H30C,&H20:NEXT I1:FOR FF=1 TO 50:NEXT FF
      FOR I1=1 TO 30:OUT &H30C,&H0:NEXT I1

REM   Check BUSYL signal indicating goniometer arm is in position.

      START=TIMER
      BUSYL=&H0

      WHILE (BUSYL AND &H1) <> &H1
          BUSYL=INP(&H30D)
          LOCATE 1,1:PRINT"SETTING GONIOMETER ANGLE"
          IF TIMER - START > 30 THEN VAR2=3:EXIT SUB
      WEND

      CLS

REM   Initiate counting (stop, reset and start scaler).

      FOR I1=1 TO 30:OUT &H30C,&H2:NEXT I1:FOR FF= 1 TO 50:NEXT FF
      FOR I1=1 TO 30:OUT &H30C,&H0:NEXT I1
      FOR I1=1 TO 60:OUT &H30C,&H4:NEXT I1:FOR FF= 1 TO 100:NEXT FF
      FOR I1=1 TO 60:OUT &H30C,&H0:NEXT I1
      FOR I1=1 TO 30:OUT &H30C,&H1:NEXT I1:FOR FF= 1 TO 50:NEXT FF
      FOR I1=1 TO 30:OUT &H30C,&H0:NEXT I1
      LOCATE 1,1:PRINT"COUNTING"

REM   Time length of count and stop scaler.

      START=TIMER

      WHILE TIMER-START < .4096*TIME
      WEND

      FOR I1=1 TO 30:OUT &H30C,&H2:NEXT I1:FOR FF=1 TO 50:NEXT FF
      FOR I1=1 TO 30:OUT &H30C,&H0:NEXT I1
      CLS

REM   Read count at base addresses 304,305,306

      F$=HEX$(NOT(INP(&H306)))
      G$=HEX$(NOT(INP(&H305)))
      H$=HEX$(NOT(INP(&H304)))
```

```
      F$=RIGHT$(F$,2):G$=RIGHT$(G$,2):H$=RIGHT$(H$,2)
      F1$=F$+G$+H$
      W=VAL(F1$)

REM   Implement check for possible zero count.
REM   If count is zero, set VAR2=4 and exit subprogram.
REM   Bell will sound and error message will be printed.

      IF W=0 THEN VAR2=4:EXIT SUB

REM   Normal exit out of Drive :

      RETURN

      END SUB


REM   Subprogram Xrdios for input and output of data.


      SUB Xrdios(J) STATIC

      DIM KI(4,15),SN$(34),KQ(16),INST$(20),FILTER$(10)

      ON J GOTO 110,120,130,140,150,160,170

REM   Open XRD1.DAT for reading of program variables.

110   OPEN "XRD1.DAT" FOR INPUT AS #1

REM   INPUT CONST(1,4) : 2 ASC const., beam/sample offset, angle incr.

      FOR I = 1 TO 4
          INPUT #1,CONST(I)
      NEXT I

REM   INPUT ANG(1-20) : 20 possible diffraction angles.

      FOR I = 1 TO 20
          INPUT #1,ANG(I)
      NEXT I

REM   INPUT VAR3(1-4) : Standard quartz masses for four lines.

      FOR I = 1 TO 4
          INPUT #1,VAR3(I)
      NEXT I

REM   INPUT KQ(1-16):for other minerals, quartz intensity equivalents.
```

```
      FOR I = 1 TO 16
          INPUT #1,KQ(I)
      NEXT I

REM   INPUT KI(1-4,1-15) : instrument calibration factors.

      FOR I = 1 TO 15

          FOR J = 1 TO 4
              INPUT #1,KI(J,I)
          NEXT J

      NEXT I

REM   INPUT EQUIVQ(1-20) : quartz line comparison to other minerals.

      FOR I = 1 TO 20
          INPUT #1,EQUIVQ(I)
      NEXT I

REM   CLOSE file XRD1.DAT

      CLOSE #1

REM   Open file XRD2.DAT containing area of last standard quartz peaks
REM   and the batch number .

      OPEN "XRD2.DAT" FOR INPUT AS #1

REM   Read standard peak values.

      FOR I = 1 TO 4
          INPUT #1,AREA(I)
      NEXT I

REM   Read batch number.

      INPUT #1,NB

REM   Close XRD2.DAT

      CLOSE #1
      SW7=INP(&H300)

      IF (SW7 AND 64) = 0 THEN

          OPEN "XRD3.DAT" FOR OUTPUT AS #1

          INPUT"Enter instrument # : ",INSTNO : PRINT #1, INSTNO
          INPUT"Enter filter type  : ",VAR2 : PRINT #1, VAR2
          INPUT"Enter number of samples : ",SAMPTOT : PRINT #1, SAMPTOT
          INPUT"Enter number of lines   : ",TOTLIN : PRINT #1, TOTLIN
```

```
          CLS
          PRINT"Enter order of lines"

          FOR J=1 TO TOTLIN
              PRINT"Enter line (";J;") :"
              INPUT"->",LIN(J) : PRINT #1, LIN(J)
          NEXT J

          CLS

REM   CHECK SW#5 AND SW#6 FOR ENHANCED PROBABILITY AND WIDE PEAK OPTION
REM   SW#5 UP ENHANCED PROBABILITY AND SW#6 UP FOR WIDE PEAK OPTION

          SW5=INP(&H300)
          SW6=SW5
          IF (SW5 AND 16) = 16 THEN PROB=4 ELSE PROB=2
          IF (SW6 AND 32) = 32 THEN PWIDTH =6 ELSE PWIDTH=4
          CLS

          PRINT #1,PROB,PWIDTH

          INPUT"Enter Company name : ",CO$ : PRINT #1, CO$
          INPUT"Enter date received: ",DATER$ : PRINT #1, DATER$
          INPUT"Our file number    : ",FIL$ : PRINT #1, FIL$

          FOR L=1 TO 4
             X=1.00
             IF INSTNO > 0 THEN X=KI(L,INSTNO)
             STDQM(L)=VAR3(L)*X
          NEXT L

          FOR L=1 TO TOTLIN
              RUNLIN=LIN(L)

              IF (RUNLIN-4) > 0 THEN
                  L3 = EQUIVQ(RUNLIN)
                  X = KQ(RUNLIN-4)
                  STDQM(RUNLIN)=STDQM(L3)*X
              END IF

          NEXT L

          FOR K=1 TO SAMPTOT
              SN$(K)=" "
              W=99.99
              CLS
              PRINT"Filter #";K;" :"
              INPUT"Enter sample number : ",SN$(K)
              INPUT"Enter sample mass   : ",W
              IF W <= 0 THEN W=.01
              MASSOUT(4,K)=W
          NEXT K
```

```
    LPRINT"SAMPLE #","SAMPLE ID","RESP.MASS"

    FOR FF=1 TO SAMPTOT
        LPRINT FF,SN$(FF),
        LPRINT USING"##.##";MASSOUT(4,FF)
    NEXT FF

    LPRINT
    LPRINT

    CLS
    INPUT"Would you like to correct an entry ? (y/n) ",ANS$
    CLS

    WHILE ANS$="y" OR ANS$="Y"
        INPUT"Enter filter number that needs correction : ",K
        CLS
        PRINT"Filter # ";K;" presently down as : "
        PRINT
        PRINT"Sample # : ";SN$(K)
        PRINT"Mass     : ";MASSOUT(4,K)
        PRINT
        INPUT"Change sample number to : ",SN$(K)
        INPUT"Change sample mass to   : ",W
        IF W <= 0 THEN W=.01
        MASSOUT(4,K)=W
        CLS
        INPUT"Are there any more corrections ? (y/n) ",ANS$
        CLS
    WEND

    FOR I= 1 TO SAMPTOT
        WRITE #1, SN$(I), MASSOUT(4,I)
    NEXT I

    CLOSE #1

ELSE

    OPEN "XRD3.DAT" FOR INPUT AS #1

    INPUT #1, INSTNO, VAR2, SAMPTOT, TOTLIN

    FOR I = 1 TO TOTLIN
        INPUT #1, LIN(I)
    NEXT I

    INPUT #1, PROB, PWIDTH
    INPUT #1, CO$,DATER$,FIL$

    FOR I = 1 TO SAMPTOT
        INPUT #1, SN$(I), MASSOUT(4,I)
    NEXT I
```

```
        FOR L=1 TO 4
            X=1.00
            IF INSTNO > 0 THEN X=KI(L,INSTNO)
            STDQM(L)=VAR3(L)*X
        NEXT L

        FOR L=1 TO TOTLIN
            RUNLIN=LIN(L)

            IF (RUNLIN-4) > 0 THEN
                L3 = EQUIVQ(RUNLIN)
                X = KQ(RUNLIN-4)
                STDQM(RUNLIN)=STDQM(L3)*X
            END IF

        NEXT L

        CLOSE #1

    END IF

    TEMPO=VAR2

    LPRINT
    LPRINT"Company        : ";CO$
    LPRINT"Date received  : ";DATER$
    LPRINT"File #         : ";FIL$
    LPRINT
    LPRINT "SAMPLE #","SAMPLE ID","RESP. MASS"
    LPRINT
    LPRINT"-----------------------------------------------------";
    LPRINT"--------------------------"
    LPRINT
    LPRINT"STANDARD ANALYSIS"
    LPRINT

    EXIT SUB

REM  This portion provides the standard analysis output.

120  GOSUB Ldata
    LPRINT"Area of standard last run = ";AREA(RUNLIN)
    LPRINT"Standard quartz mass = ";VAR3(RUNLIN)
    LPRINT
    LPRINT
    EXIT SUB

REM  This section gives diagnostic print-out (no agreement).
```

```
130  GOSUB Ldata
     GOSUB Plot
     EXIT SUB

REM  This is another diagnostic print-out.

140  LPRINT "FILTER # = ";SAMPNO
     LPRINT "TIME = ";TIME
     LPRINT "J = ";J
     LPRINT

     FOR K=1 TO 29
         LPRINT COUNT(K);
     NEXT K

     LPRINT
     GOSUB Plot
     EXIT SUB

REM  This section provides the normal sample output.

150  GOSUB Samres
     EXIT SUB

REM  This section provides the normal final table.

160  NB=NB+1

     LPRINT
     LPRINT"Company        : ";CO$
     LPRINT"Date received  : ";DATER$
     LPRINT"File #         : ";FIL$
     LPRINT
     LPRINT"-------------------------------------------------------";
     LPRINT"--------------------------"
     LPRINT
     LPRINT "BATCH NUMBER = ";NB
     LPRINT
     INST$(1)="SAMPLER : CAMPEDS (BMRC)  25mm filter at 2.0 L/min."
     INST$(2)="SAMPLER : CAMPEDS (ACGIH) 25mm filter at 2.5 L/min."
     INST$(3)="SAMPLER : ----------------------------------------"
     INST$(4)="SAMPLER : ----------------------------------------"
     INST$(5)="SAMPLER : Metal Cyclones (BMRC) 25mm filter at 1.9 L/min."
     INST$(6)="SAMPLER : Metal Cyclones (AIH) 25mm filter at 2.4 L/min."
     INST$(7)="SAMPLER : MRE-113A (3 chan.) 25mm filter at 1.875 L/min."
     INST$(8)="SAMPLER : YORK Cyclone (SM0S) 25mm filter 1.7 L/min."
     INST$(9)="SAMPLER : YORK Cyclone (SP3S) 25mm filter 1.7 L/min."
     INST$(10)="SAMPLER : YORK Cyclone (NP2S) 25mm filter 1.7 L/min."
     INST$(11)="SAMPLER : YORK Cyclone (SP3S) 37mm filter 1.7 L/min."
     INST$(12)="SAMPLER : YORK Cyclone (NP3S) 37mm filter 1.7 L/min."
     INST$(13)="SAMPLER : YORK Cyclone (NP3S) 25mm filter 1.7 L/min."
     INST$(14)="SAMPLER : ----------------------------------------"
     INST$(15)="SAMPLER : SIMPEDS 25mm filter "
     INST$(16)="SAMPLER : SIMPEDS 25mm filter "
```

```
      FILTER$(1)="FILTER : 5um Silver Membrane"
      FILTER$(2)="FILTER : 3um Silver Membrane"
      FILTER$(3)="FILTER : 1.2um Silver Membrane"
      FILTER$(4)="FILTER : 0.8um Silver Membrane"
      FILTER$(5)="FILTER : Vinyl"

      LPRINT INST$(INSTNO+1)
      LPRINT FILTER$(TEMPO+1)
      IF LMC = 1 THEN LPRINT"LIGHT MASS CORRECTION IN EFFECT"
      LPRINT "NUMBER OF SAMPLES : ";SAMPNO
      LPRINT

      Y2=2000
      Y0=-2000

      GOSUB Fintab

      OPEN "XRD2.DAT" FOR OUTPUT AS #1

      FOR K = 1 TO 4
          PRINT #1,AREA(K);
      NEXT K

      PRINT #1,NB
      CLOSE #1

      GOSUB Absorp

      LPRINT"ABSORPTION CORRECTED"
      LPRINT"---------- ---------"
      LPRINT

      GOSUB Fintab
      EXIT SUB

REM   This section provides beep if a mistake is made during set-up.

170   SOUND 2000,200
      LPRINT
      LPRINT"X-RAY OR INTERFACE FAULT"
      EXIT SUB

REM   Subroutine Ldata for detailed line output.

      Ldata:

      LPRINT"Line : ";RUNLIN
      LPRINT"------"
      LPRINT
      LPRINT "COUNTS = ";
```

```
      FOR K= 1 TO 29
          LPRINT COUNT(K);
      NEXT K

      LPRINT
      LPRINT "PEAK POSITIONS = ";

      FOR K = 1 TO 5
          LPRINT PEAK(K);
      NEXT K

      LPRINT
      LPRINT"AREA = ";AREA
      RETURN

REM   Subroutine Samres

      Samres:

      FOR K = 1 TO 2
          X=VAR3(K)
          Y=-1000
          IF X > 0 THEN Y=VAR3(K+2)/X
          MASSOUT(K,SAMPNO)=Y
      NEXT K

      LPRINT SAMPNO;" SAMPLE # ";SN$(SAMPNO);" TOT. RESP. MASS = ";
      LPRINT MASSOUT(4,SAMPNO);" mg. ";" RESP. QUARTZ = ";
      LPRINT CINT(MASSOUT(1,SAMPNO));" ug. "
      LPRINT
      LPRINT "LINE #","CNT TIME","PEAK POS","MASS"
      LPRINT

      FOR K = 1 TO TOTLIN
          LPRINT LINEOUT(K),TIMEOUT(K),PEAKOUT(K),QMASSA(K)
      NEXT K

      LPRINT
      LPRINT"------------------------------------------------------";
      LPRINT"--------------------------"
      LPRINT
      RETURN

REM   Subroutine used to plot peaks.

      Plot:

      MAXCNT=0
      MINCNT=1000000
```

```
      FOR N=1 TO 29
          W=COUNT(N)

          IF W > 0 THEN
              IF (W-MAXCNT)  > 0 THEN MAXCNT = W
              IF (W-MINCNT)  < 0 THEN MINCNT = W
          END IF

      NEXT N

      IF (MAXCNT-2*MINCNT) < 0 THEN MAXCNT = 2*MINCNT

      LPRINT"MINIMUM COUNT = ";MINCNT;" MAXIMUM COUNT = ";MAXCNT

      FOR N = 1 TO 29
          W=COUNT(N)

          IF W>0 THEN
              K=(W-MINCNT)/(MAXCNT-MINCNT)*70+1
              LPRINT CHR$(64+N);

              FOR NN=1 TO K
                  LPRINT" ";
              NEXT NN

              LPRINT "*"
          END IF

      NEXT N

      RETURN

REM   Subroutine Fintab

      Fintab:

      LPRINT
      LPRINT "SAMPLE #    ";"TOTAL(mg)   ";"QUARTZ(mg) ";"    %        ";
      LPRINT "CRYST.(mg) ";"     %       ";"SILVER      "
      LPRINT

      FOR K = 1 TO SAMPNO
          W=MASSOUT(4,K)
          X1=MASSOUT(1,K)
          X2=MASSOUT(2,K)
          Y1=MASSOUT(3,K)

          IF (Y0-Y2-Y1) > 0 THEN
              Y1=SQR(Y0/Y1)
              X1=X1*Y1
              X2=X2*Y1
          END IF
```

```
        X=X1/W/10
        Y=X2/W/10
        LPRINT USING "\     \"; SN$(K);
        LPRINT USING "########.##";W;
        LPRINT USING "#######.###";X1/1000;
        LPRINT USING "########.";X;
        LPRINT USING "#############.";X2;
        LPRINT USING "########.";Y;
        LPRINT USING "#########.##";Y1
    NEXT K

    LPRINT
    LPRINT
    LPRINT
    LPRINT
    LPRINT
    LPRINT
    LPRINT
    RETURN

REM  Subroutine Absorp. for silver peak absorption correction.

    Absorp:

    X1=0:X2=0:Y1=0:Y2=0

    FOR K=1 TO SAMPNO
        Y=MASSOUT(3,K)
        IF Y <= 0 THEN EXIT SUB
        W=EXP(MASSOUT(4,K))
        X1=X1+W
        X2=X2+W*W
        Y1=Y1+Y
        Y2=Y2+Y*Y
        XY=XY+W*Y
    NEXT K

    X=SAMPNO
    XY=XY-X1*Y1/X
    X2=X2-X1*X1/X
    Y2=Y2-Y1*Y1/X
    Y0=Y1/X+XY/X2*(1-X1/X)
    Y2=SQR((1-XY*XY/X2/Y2)*Y2/(X-2.0))

    RETURN

    END SUB
```

APPENDIX B

This appendix lists the Microsoft Quickbasic code for the program XRDDATPR.BAS.

```
REM  XRDDATPR.BAS
REM  XRDDATPR displays the contents of the XRD1.DAT file
REM       and allows changes to be made.

     OPTION BASE 1

     DIM menu1$(8), menu2$(8), const(4), angle(20), qStdM(4)
     DIM kalFact(4,15), linComp(20), qLinConv(16)

     COMMON SHARED  linComp(), KalFact()
     COMMON SHARED  const(), angle(), qStM(), qLinConv()

     CLS
     LOCATE 3,30:PRINT"Program XRDDATPR.EXE"
     CALL Frame(13,70,5,16)

     LOCATE 7,15:PRINT"XRDDATPR.EXE is software used to examine and/or m
     LOCATE 8,15:PRINT"the XRD1.DAT data file used in conjunction with t
     LOCATE 9,15:PRINT"X-RAY  diffraction program  XRD. This software is
     LOCATE 10,15:PRINT"driven. Two menus are available, the first  one
     LOCATE 11,15:PRINT"the user to examine the data file's content, whi
     LOCATE 12,15:PRINT"the second one performs changes to the  existing
     LOCATE 14,15:PRINT"Press <ENTER> to initiate program."

     CALL Pause
     CLS

     REM $INCLUDE : 'UPPER.BAS'

     FOR i% = 1 TO 8
         READ menu1$(i%)
         READ menu2$(i%)
     NEXT i%

     DATA constants, constants
     DATA angles, angles
     DATA standard quartz mass, standard quartz mass
     DATA line correction factors, line correction factors
     DATA instrument correction factors, instrument correction factors
     DATA quartz line comparison, quartz line comparison
     DATA new data entry menu, display menu
     DATA exit program, exit program


REM  READ from XRD1.DAT

     OPEN "XRD1.DAT" FOR INPUT AS # 1
```

```
      FOR i% = 1 TO 4
          INPUT #1, const(i%)
      NEXT i%

      FOR i% = 1 TO 20
          INPUT #1, angle(i%)
      NEXT i%

      FOR i% = 1 TO 4
          INPUT #1, qStdM(i%)
      NEXT i%

      FOR i% = 1 TO 16
          INPUT #1, qLinConv(i%)
      NEXT i%

      FOR i% = 1 TO 15
          FOR j% = 1 TO 4
              INPUT #1, kalFact(j%, i%)
          NEXT j%
      NEXT i%

      FOR i% = 1 TO 20
          INPUT #1, linComp(i%)
      NEXT i%

      CLOSE #1

REM   Show display menu1$, get user's choice and go to appropriate sectio

101   choice%=0

      WHILE choice% <> 7
          LOCATE 2,27: PRINT"XRD1.DAT Data File Display"
          CALL Menu(menu1$(),choice%)

          IF (choice% = 1) THEN
              CALL dispConst("Constants used in XRD program")
          ELSEIF (choice% = 2) THEN
              CALL dispAngles("Angles used in XRD program")
          ELSEIF (choice% = 3) THEN
              CALL dispqStdM("Standard's equivalent quartz mass")
          ELSEIF (choice% = 4) THEN
              CALL dispqLinConv("Quartz line equivalent intensities")
          ELSEIF (choice% = 5) THEN
              CALL dispkalFact("Instrument calibration factors")
          ELSEIF (choice% = 6) THEN
              CALL displinComp("Quartz line comparison scheme")
```

```
        ELSEIF (choice% = 8) THEN
            CALL Fildat
            END
        END IF

    WEND

REM  Show Data modification menu

    choice%=0

    WHILE choice% <> 7
        LOCATE 2,25:PRINT"XRD1.DAT File modification menu"
        CALL Menu(menu2$(),choice%)

        IF (choice% = 1) THEN
            CALL modConst
        ELSEIF (choice% = 2) THEN
            CALL modAngles
        ELSEIF (choice% = 3) THEN
            CALL modqStdM
        ELSEIF (choice% = 4) THEN
            CALL modqLinConv
        ELSEIF (choice% = 5) THEN
            CALL modkalFact
        ELSEIF (choice% = 6) THEN
            CALL modlinComp
        ELSEIF (choice% = 8) THEN
            CALL Fildat
            END
        END IF

    WEND

    GO TO 101

    END

REM  Subprograms

REM  $INCLUDE : 'FRAME.BAS'
REM  $INCLUDE : 'MENU.BAS'

    SUB dispConst(mString$) STATIC

    lc%=((80-LEN(mString$))\2)-3
    rc%=80-lc%
    tc%=7 : bc%=16
    CALL Frame(lc%,rc%,tc%,bc%)
    LOCATE tc%+2,lc%+4:PRINT mString$
    LOCATE tc%+4,lc%+4:PRINT"ASC constant #1 ";const(1)
```

```
LOCATE tc%+5,lc%+4:PRINT"ASC constant #2 ";const(2)
LOCATE tc%+6,lc%+4:PRINT"Beam/Filter disp. ";const(3)
LOCATE tc%+7,lc%+4:PRINT"Angle increment ";const(4)
LOCATE 18,lc%+4:PRINT"Hit <ENTER> to continue"
CALL Pause
CLS

END SUB

SUB dispAngles(mString$) STATIC

lc%=13 : rc%=64
tc%=7 : bc%=17
CALL Frame(lc%,rc%,tc%,bc%)
LOCATE tc%+2,lc%+9:PRINT mString$
LOCATE tc%+4,lc%+4:PRINT"quartz       ";angle(1);" "_
;angle(2);" ";angle(3);" ";angle(4)
LOCATE tc%+5,lc%+4:PRINT"crystobalite ";angle(5);" "_
;angle(6);" ";angle(7);" ";angle(8)
LOCATE tc%+6,lc%+4:PRINT"tridymite    ";angle(9);" "_
;angle(10);" ";angle(11);" ";angle(12)
LOCATE tc%+7,lc%+4:PRINT"not used     ";angle(13);" "_
;angle(14);" ";angle(15);" ";angle(16)
LOCATE tc%+8,lc%+4:PRINT"sliver(last) ";angle(17);" "_
;angle(18);" ";angle(19);" ";angle(20)
LOCATE 19,lc%+4:PRINT "Hit <ENTER> to continue"
CALL Pause
CLS

END SUB

SUB dispqStdM(mString$) STATIC
lc%=20 : rc%=60
tc%=8  : bc%=14
CALL Frame(lc%,rc%,tc%,bc%)
LOCATE tc%+2,lc%+4:PRINT mString$
LOCATE tc%+4,lc%+4:PRINT qStdM(1);" ";qStdM(2);" ";_
qStdM(3);" ";qStdM(4)
LOCATE 16,lc%+4:PRINT"Hit <ENTER> to continue"
CALL Pause
CLS

END SUB

SUB dispqLinConv(mString$) STATIC

lc%=14:rc%=66
tc%=7:bc%=16
CALL Frame(lc%,rc%,tc%,bc%)
LOCATE tc%+2,lc%+4:PRINT mString$
```

```
LOCATE tc%+4,lc%+4:PRINT"crystobalite ";qLinConv(1);" "-
;qLinConv(2);" ";qLinConv(3);" ";qLinConv(4)
LOCATE tc%+5,lc%+4:PRINT"tridymite    ";qLinConv(5);" "_
;qLinConv(6);" ";qLinConv(7);" ";qLinConv(8)
LOCATE tc%+6,lc%+4:PRINT"not used     ";qLinConv(9);" ";_
qLinConv(10);" ";qLinConv(11);" ";qLinConv(12)
LOCATE tc%+7,lc%+4:PRINT"silver(last) ";qLinConv(13);" ";_
qLinConv(14);" ";qLinConv(15);" ";qLinConv(16)
LOCATE 18,lc%+4:PRINT"Hit <ENTER> to continue"
CALL Pause
CLS

END SUB

SUB dispkalFact(mString$) STATIC
lc%=10:rc%=70
tc%=2:bc%=22
CALL Frame(lc%,rc%,tc%,bc%)
LOCATE tc%+2,lc%+4:PRINT mString$

FOR i% = 1 TO 15

    IF i%<=9 THEN
        LOCATE tc%+3+i%,lc%+4:PRINT"instrument #  ";i%;" "_
        ;kalFact(1,i%);" ";kalFact(2,i%);" ";kalFact(3,i%);_
        " ";kalFact(4,i%)

    ELSE

        LOCATE tc%+3+i%,lc%+4:PRINT"instrument # ";i%;" "_
        ;kalFact(1,i%);" ";kalFact(2,i%);" ";kalFact(3,i%);_
        " ";kalFact(4,i%)

    END IF

NEXT i%

LOCATE 24,lc%+4:PRINT"Hit <ENTER> to continue"
CALL Pause
CLS

END SUB

SUB displinComp (mString$) STATIC

lc%=22:rc%=58
tc%=7:bc%=17
CALL Frame(lc%,rc%,tc%,bc%)
LOCATE tc%+2,lc%+4:PRINT mString$
LOCATE tc%+4,lc%+4:PRINT"quartz       ";linComp(1);" "_
;linComp(2);" ";linComp(3);" ";linComp(4)
```

```
LOCATE tc%+5,lc%+4:PRINT"crystobalite ";linComp(5);" "_
;linComp(6);" ";linComp(7);" ";linComp(8)
LOCATE tc%+6,lc%+4:PRINT"tridymite    ";linComp(9);" "_
;linComp(10);_ " ";linComp(11);" ";linComp(12)
LOCATE tc%+7,lc%+4:PRINT"not used     ";linComp(13);" "_
;linComp(14);" ";linComp(15);" ";linComp(16)
LOCATE tc%+8,lc%+4:PRINT"sliver(last) ";linComp(17);" "_
;linComp(18);" ";linComp(19);" ";linComp(20)
LOCATE 19,lc%+4:PRINT"Hit <ENTER> to continue"
CALL Pause
CLS

END SUB

SUB modConst STATIC

PRINT"Constants are : ";const(1),const(2),const(3),const(4)
PRINT""

INPUT"Enter ASC(1), ASC(2), axis corr. and angle incr. "_
,const(1),_ const(2),const(3),const(4)
CLS

END SUB

SUB modAngles STATIC

INPUT"Enter range of angles to change (eg. 1,4 to change _
quartz angles)",k%,j% CLS

FOR i% = k% TO j%
    PRINT"Angle ";i%;" is now ";angle(i%)
    INPUT"New angle is : ", angle(i%)
NEXT i%

CLS

END SUB

SUB modqStdM STATIC

PRINT"Standard masses are: ";qStdM(1);" ";qStdM(2);" "_
;qStdM(3);" ";qStdM(4) PRINT " "
INPUT"Enter new standard masses : "-
,qStdM(1),qStdM(2),qStdM(3),qStdM(4)
CLS

END SUB
```

```
SUB modqLinConv STATIC
INPUT"Enter 0-crysto. 1-tridym. 2-not used 3-silver ";j%
CLS

FOR i% = 4*j%+1 TO 4*j%+4
    PRINT"Line conv. factor #  ";i%;" is now : ";qLinConv(i%)
    PRINT" "
    PRINT"Enter new line conv. factor to quartz # "; i%
    INPUT" -> ", qLinConv(i%)
    CLS
NEXT i%

CLS

END SUB

SUB modkalFact STATIC

INPUT "Enter instrument number (1-15) ",i%
CLS

FOR j% = 1 TO 4
    PRINT"Instrument #";i%;" cal. fact. line  #";j%;" is now _
    : ";kalFact(j%,i%)
    PRINT"Enter new instrument calibration factor # ";j%
    INPUT" -> ", kalFact(j%,i%)
    CLS
NEXT j%

CLS

END SUB

SUB modlinComp STATIC

INPUT"Enter 0-quartz 1-crysto. 2-tridym. 3-not used 4-silver _
";j%
CLS

FOR i% = 4*j%+1 TO 4*j%+4
    PRINT"Quartz comp. line for line #";i%;" is now : _
    ";linComp(i%)
    PRINT"Enter new quartz comp. line scheme line # ";i%
    INPUT" -> ", linComp(i%)
    CLS
NEXT i%

CLS

END SUB
```

```
SUB Fildat STATIC

OPEN "XRD1.DAT" FOR OUTPUT AS #1

FOR i% = 1 TO 4
    WRITE #1,const(i%)
NEXT i%

FOR i% = 1 TO 20
    WRITE #1,angle(i%)
NEXT i%

FOR i% = 1 TO 4
    WRITE #1,qStdM(i%)
NEXT i%

FOR i% = 1 TO 16
    WRITE #1,qLinConv(i%)
NEXT i%

FOR i%= 1 TO 15

    FOR j% = 1 TO 4
        WRITE #1,kalFact(j%,i%)
    NEXT j%

NEXT i%

FOR i% = 1 TO 20
   WRITE #1, linComp(i%)
NEXT i%

CLOSE #1

END SUB

SUB Pause STATIC

character$=""

WHILE character$=""
    character$=INKEY$
WEND

END SUB
```